

ADT PLM

Programmer's Learning Machine

Matthieu Nicolas

COAST meeting, 2016-02-12

1 Presentation of PLM

- Purposes
- Demo
- About PLM
- Architecture
- Targets

2 To a web app

- Goals
- Server-side
- Client-side

3 Assessment of user's code

- Challenges
- Extraction of the execution component

4 Result

5 Current tasks

1 Presentation of PLM

- Purposes
- Demo
- About PLM
- Architecture
- Targets

2 To a web app

- Goals
- Server-side
- Client-side

3 Assessment of user's code

- Challenges
- Extraction of the execution component

4 Result

5 Current tasks

Presentation of PLM

Purposes

- Application to learn programming.

Presentation of PLM

Purposes

- Application to learn programming.
- Allows students to progress at their own speed...

Presentation of PLM

Purposes

- Application to learn programming.
- Allows students to progress at their own speed...
- ... while the teacher helps the ones having trouble.

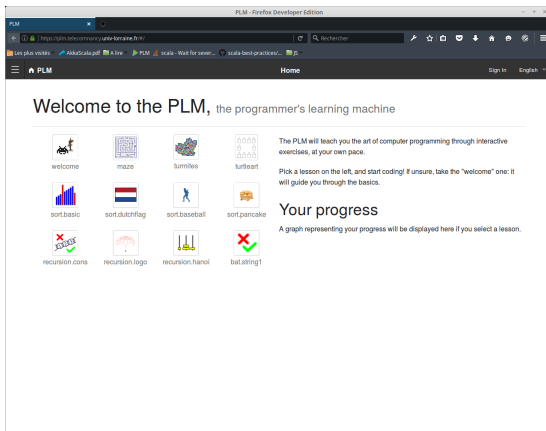
Presentation of PLM

Purposes

- Application to learn programming.
- Allows students to progress at their own speed...
- ... while the teacher helps the ones having trouble.
- Used at TELECOM Nancy since 2008.

Presentation of PLM

Quick demo



- Available at <https://plm.telecomnancy.univ-lorraine.fr>

Presentation of PLM

12 lessons, 200 exercises

PLM - Mozilla Firefox

https://plm.univ-lorraine.fr/la/lessons/maze/mazeWallFollowerWallFollowerMaze

Se connecter Java Français

Maze / WallFollowerMaze

World Objective



Résultats de l'exécution

Éditeur de code:

```
public void run() {  
    // ...  
}
```

Cas de tests:

Another labyrinth Labyrinth

Presentation of PLM

12 lessons, 200 exercises

PLM - Mozilla Firefox

https://plm.univ-lorraine.fr/.../recursion/hanoi/hanoi.lessons/hanoi.Hani

Rechercher

Les plus visités Linux Mint Community Forums Blog News NinjaMock - free tool... Spotify Web Player Wau The Most Amaz... Coursa P2P

PLM Recursion.hanoi / HanoiBoard Se connecter Java Français

World Objective

0 Move



Résultats de l'exécution

Cas de tests:

solve(0,1,2) solve(1,2,0) solve(2,0,1)

Éditeur de code: Réinitialiser API

```
public void hanoi(int height, int src, int other, int dest) {  
}
```

Presentation of PLM

12 lessons, 200 exercises

PLM - Mozilla Firefox

https://plm.univ-lorraine.fr/ta/lessons/recursion/logo/recursion/logo/tree.Tre

Les plus visités Linux Mint Community Forums Blog News NinjaMock - free tool... Spotify Web Player Wau The Most Amaz... Coursa P2P

PLM Recursion.logo / Tree Se connecter Java Français

World Objective



Éditeur de code: Réinitialiser API

```
public void tree(int steps, double length, double angle, double shrink) {  
    // ...  
}
```

Résultats de l'exécution

Cas de tests:

tree(10,100,90,0.68)	▶	tree(10,80,45,0.7)	▶
tree(7,75,15,0.8)	▶	tree(7,75,30,0.8)	▶

Presentation of PLM

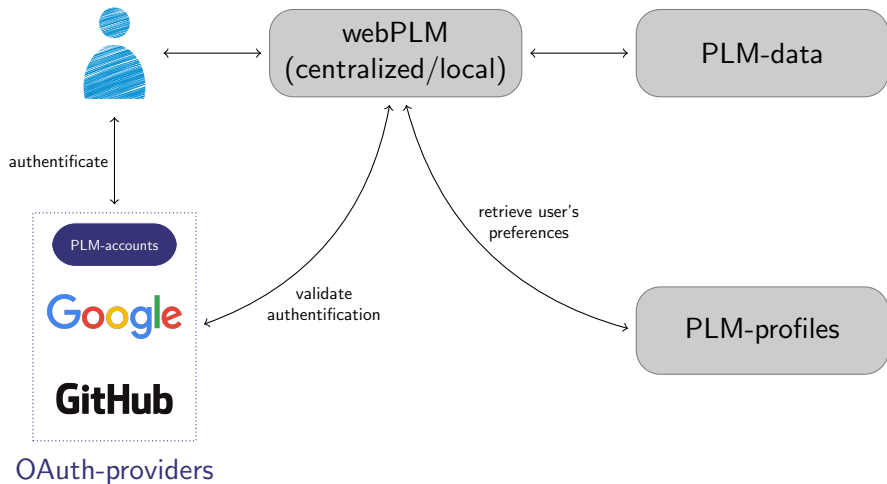
Languages and programming languages

- Available languages:
 - English
 - French
 - Brazilian Portuguese
- Supported programming languages:



Presentation of PLM

Application's architecture



Presentation of PLM

A word about PLM-data

- Keep track of the users' progress...

Presentation of PLM

A word about PLM-data

- Keep track of the users' progress...
- ... using a git repository



Presentation of PLM

How does it work?

- Store users' code versions

Presentation of PLM

How does it work?

- Store users' code versions
- Store users' actions as commit messages



```
1 {  
2   · "kind" : "executed",  
3   · "lang" : "Java",  
4   · "exo" : "welcome.lessons.instructions.Instructions",  
5   · "passedtests" : 1,  
6   · "totaltests" : 1,  
7   · "outcome" : "pass"  
8 }
```

Presentation of PLM

How does it work?

- Store users' code versions
- Store users' actions as commit messages



```
1 {  
2   · "kind" : "executed",  
3   · "lang" : "Java",  
4   · "exo" : "welcome.lessons.instructions.Instructions",  
5   · "passedtests" : 1,  
6   · "totaltests" : 1,  
7   · "outcome" : "pass"  
8 }
```

- Working in anonymous branches
- Branches pushed to a **GitHub** repo

Presentation of PLM

Targets

- Students obviously
 - More teaching content
 - Gamification

Presentation of PLM

Targets

- Students obviously
 - More teaching content
 - Gamification
- But also teachers
 - Keep track of the students' progress
 - Adapt content to their needs
 - Able to add their own exercises

Presentation of PLM

Targets

- Students obviously
 - More teaching content
 - Gamification
- But also teachers
 - Keep track of the students' progress
 - Adapt content to their needs
 - Able to add their own exercises
- And researchers
 - To an experimental teaching platform
 - How to detect students having difficulties?
 - What are the most common errors?

Outline

1 Presentation of PLM

- Purposes
- Demo
- About PLM
- Architecture
- Targets

2 To a web app

- Goals
- Server-side
- Client-side

3 Assessment of user's code

- Challenges
- Extraction of the execution component

4 Result

5 Current tasks

To a web app

Evolution of the project

- Formerly a fat client
 - Written in Java

To a web app

Evolution of the project

- Formerly a fat client
 - Written in Java
- Switch to a web application
 - Server implemented in Scala using *PlayFramework*
 - User interface written in Javascript using *AngularJS* and *Foundation*



To a web app

Motivations

- Want to switch to SaaS¹
 - Easy to use
 - Easy to update
 - Easy to track usage data
- More user-friendly
- Aim to setup SPOC² and MOOC³
- But don't have the time and means for a reboot

¹Software as a Service

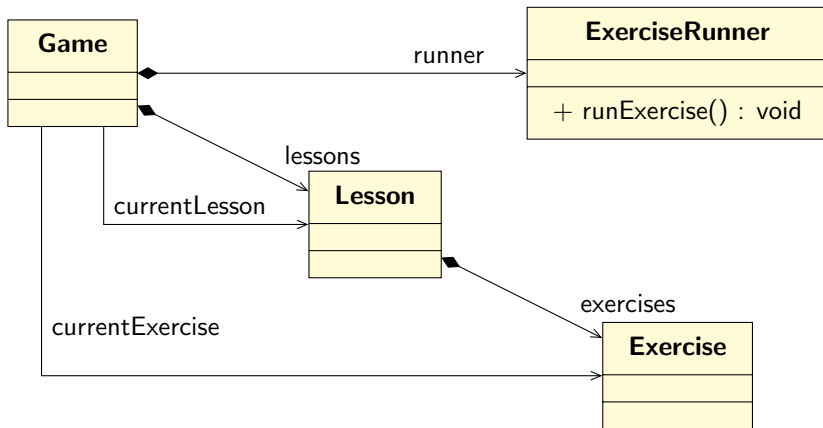
²Small Private Online Course

³Massive Open Online Course

To a web app

Refactoring PLM

- Implemented a headless version of PLM: *PLM-engine*
 - Provide all PLM's content and methods
 - But without a user interface



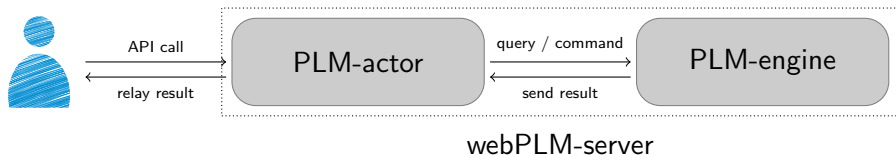
To a web app

Implementing the server

- Designed an API over PLM-engine
- Only need to implement a controller
 - Verify calls received from the client
 - Query or command PLM-engine according to the call
 - Send back result or acknowledgement to the client

To a web app

Interactions between components



To a web app

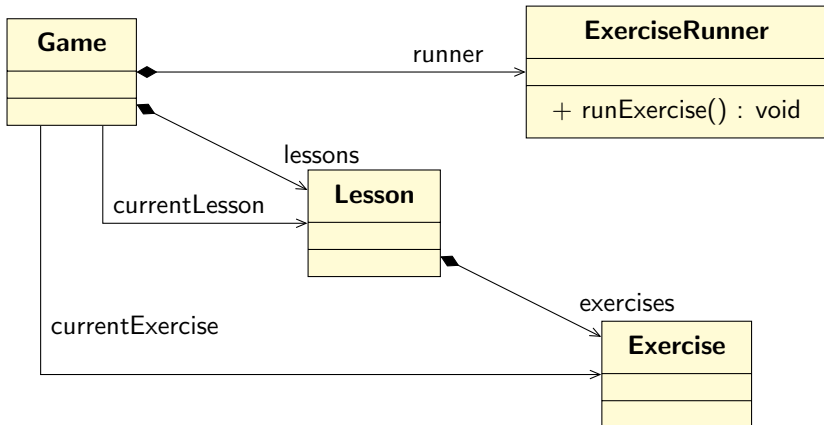
Dealing with multi-user

- **Game** is a *singleton*

To a web app

Dealing with multi-user

- **Game** is a *singleton*
- Do you remember that we store the user's session in **Game**?



To a web app

Removing the singleton **Game**

- Need to refactor all components accessing it

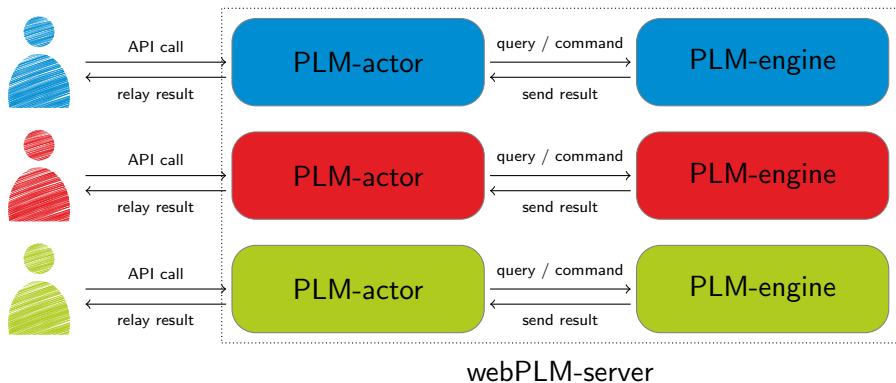
To a web app

Removing the singleton **Game**

- Need to refactor all components accessing it
- Let's save it for later!
- Add **Game** as constructor's parameter

To a web app

Multi-user scenario



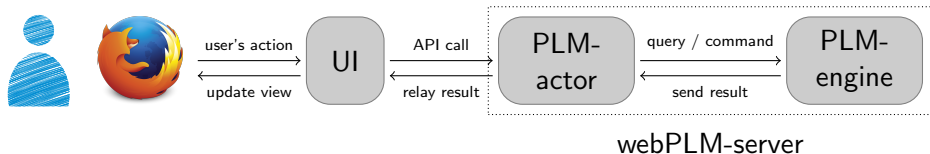
To a web app

Results

- Build quickly a web server from the fat client...
- ... but we also need a user interface

To a web app

Adding the user interface



To a web app

- Have to translate user's actions into API calls
- Have to re-implement PLM-engine's data models

Outline

1 Presentation of PLM

- Purposes
- Demo
- About PLM
- Architecture
- Targets

2 To a web app

- Goals
- Server-side
- Client-side

3 Assessment of user's code

- Challenges
- Extraction of the execution component

4 Result

5 Current tasks

Assessment of user's code

Limits

- Run on the same machine, same JVM

Assessment of user's code

Limits

- Run on the same machine, same JVM
- How to protect ourselves from users' rookie mistakes?
 - Infinite loops

Assessment of user's code

Limits

- Run on the same machine, same JVM
- How to protect ourselves from users' rookie mistakes?
 - Infinite loops
- And from more malicious "mistakes"?
 - Infinite thread creation
 - Endless file creation

Assessment of user's code

Limits

- Run on the same machine, same JVM
- How to protect ourselves from users' rookie mistakes?
 - Infinite loops
- And from more malicious "mistakes"?
 - Infinite thread creation
 - Endless file creation
- And from *System.exit(whatever)*?

Assessment of user's code

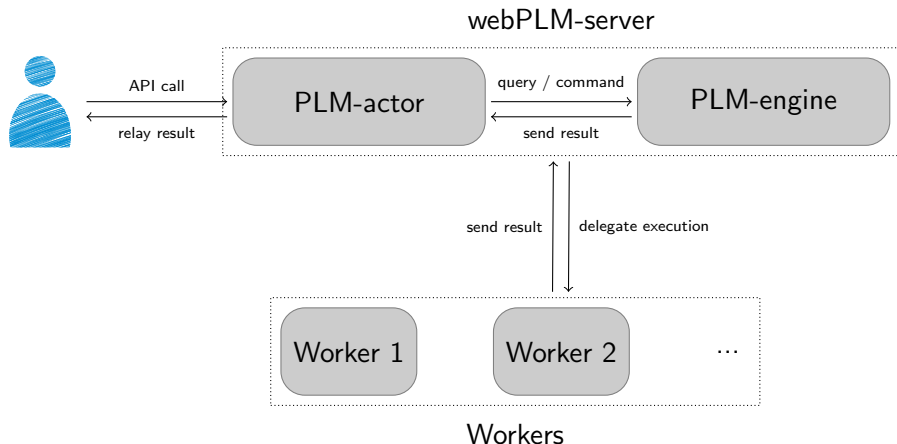
Limits

- Run on the same machine, same JVM
- How to protect ourselves from users' rookie mistakes?
 - Infinite loops
- And from more malicious "mistakes"?
 - Infinite thread creation
 - Endless file creation
- And from *System.exit(whatever)*?
- Scalability issues

Assessment of user's code

Chosen solution

- Delegate execution to workers



Assessment of user's code

The judges

- Called *Judges* in the litterature
- Use PLM-engine as well
- Workflow:
 - Retrieve an execution request
 - Parse the request to extract parameters
 - Configure PLM-engine according to them
 - Run the user's code
 - Send back result to webPLM

Assessment of user's code

Message queues

- Message-driven architecture
- Loosely coupled system
- Asynchronous/Synchronous
- Help to implement:
 - Producer/Consumer pattern
 - Request/Response pattern
- Different reliability patterns of the message processing:
 - Only one worker
 - At least one worker
 - All workers
- Easy to scale

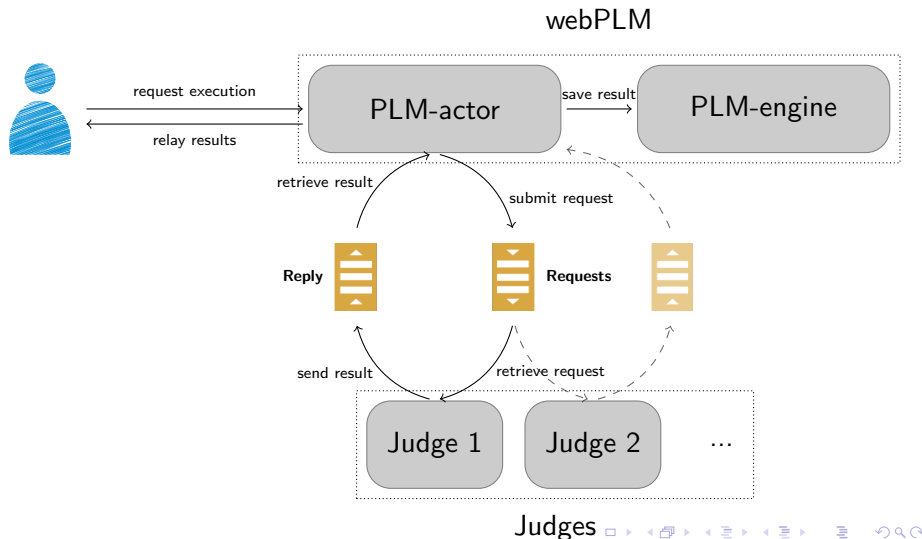
Assessment of user's code

Message queues

- Message-driven architecture
- Loosely coupled system
- **Asynchronous**/Synchronous
- Help to implement:
 - Producer/Consumer pattern
 - **Request/Response pattern**
- Different reliability patterns of the message processing:
 - **Only one worker**
 - At least one worker
 - All workers
- Easy to scale

Assessment of user's code

Architecture with judges



Assessment of user's code

Pros and cons

- Pros:
 - Allow to run code without impacting webPLM's performances
 - Meet the scalability requirements

Assessment of user's code

Pros and cons

- Pros:

- Allow to run code without impacting webPLM's performances
- Meet the scalability requirements

- Cons:

- Make sure to use the right version of PLM-engine
- Need to deploy them easily
- Should restart them after each execution
- Have to restrict their resources usage

Assessment of user's code

Docker

- Lightweight virtualization tool
- Build image of your application
- Run containers based on images



Assessment of user's code

In our case

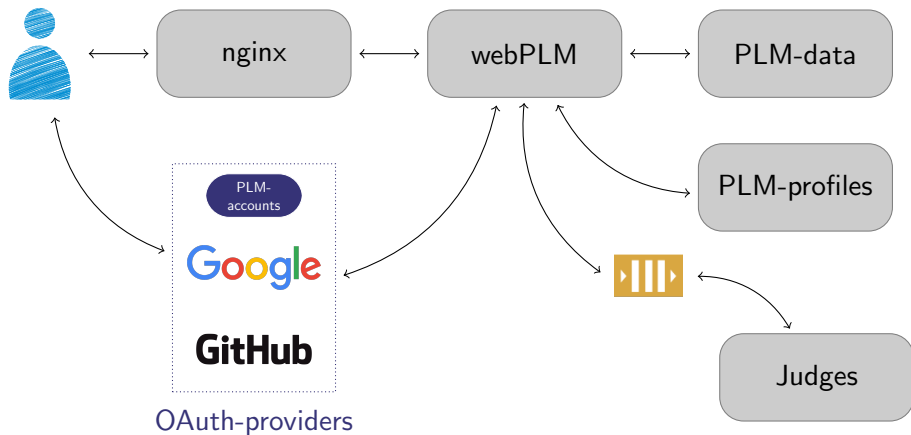
- Deploy easily all components
- Restart judges automatically
- Limit judges' ressources

Outline

- 1 Presentation of PLM
 - Purposes
 - Demo
 - About PLM
 - Architecture
 - Targets
- 2 To a web app
 - Goals
 - Server-side
 - Client-side
- 3 Assessment of user's code
 - Challenges
 - Extraction of the execution component
- 4 Result
- 5 Current tasks

Result

Current architecture



Result

Live-session in TELECOM Nancy

- Used in TELECOM Nancy in September 2015
- 30 hours of live testing with 100 students

Result

Live-session in TELECOM Nancy

- Used in TELECOM Nancy in September 2015
- 30 hours of live testing with 100 students
- Engine is (almost) working fine...
- ... but user experience needs to be improved!

- Scalability issues:
 - Work well with small exercises
 - Can't cope with workload of larger exercises

Result

Live-session in TELECOM Nancy

- Scalability issues:
 - Work well with small exercises
 - Can't cope with workload of larger exercises
- No tools for monitoring set up...

Result

Live-session in TELECOM Nancy

- Scalability issues:
 - Work well with small exercises
 - Can't cope with workload of larger exercises
- No tools for monitoring set up...
- ... so the bottleneck is unknown.

Outline

- 1 Presentation of PLM
 - Purposes
 - Demo
 - About PLM
 - Architecture
 - Targets
- 2 To a web app
 - Goals
 - Server-side
 - Client-side
- 3 Assessment of user's code
 - Challenges
 - Extraction of the execution component
- 4 Result
- 5 Current tasks

Current tasks

Refactor PLM-engine

- Want to remove lessons and exercises from PLM-engine, from **Game**
 - Need to release new version of webPLM and Judge for each change
 - Heavy and error prone workflow
 - Want to implement an exercise editor

Current tasks

Refactor PLM-engine

- Want to remove lessons and exercises from PLM-engine, from **Game**
 - Need to release new version of webPLM and Judge for each change
 - Heavy and error prone workflow
 - Want to implement an exercise editor
- Gradually shifted to get rid of **Game**
 - More annoying than useful

Current work

Rework workflow with judges

- Judges had all the exercises locally
 - Only needed to retrieve the exercise's ID

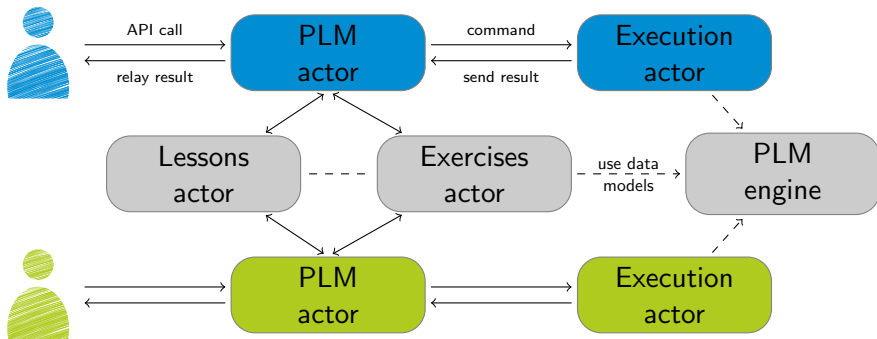
Current work

Rework workflow with judges

- Judges had all the exercises locally
 - Only needed to retrieve the exercise's ID
- Now need to transfer the exercise's model
 - Implement methods to export/import exercises as/from JSON

Current work

WIP architecture



Current tasks

Solve performance issues

- Set up some monitoring tools
- Perform some load testing to identify the bottleneck

Next steps

Sneak peek from the TODO list

- Integrate interns' contributions
- Set up Continuous Deployment
- Support additional programming languages
- Implement a debug mode similar to popular IDEs' ones
- Add features to help teachers to supervise their students
- ...

Thanks for your attention, any questions?